



# A distributed evolutionary approach for fast 3-D stereo reconstruction

Amine Boumaza

## ► To cite this version:

Amine Boumaza. A distributed evolutionary approach for fast 3-D stereo reconstruction. 2008. inria-00263542

**HAL Id: inria-00263542**

**<https://inria.hal.science/inria-00263542>**

Preprint submitted on 12 Mar 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A distributed evolutionary approach for fast 3-D stereo reconstruction

Amine. Boumaza

13/01/2008

## Abstract

This article describes the fly algorithm an evolutionary approach aimed at fast three-dimensional scene reconstruction from stereo images. We describe the main component of the algorithm and present an experimental analysis of its results on benchmark data, showing that a good results can be obtained at very low computational cost. We hope through this work motivate the use of such algorithms in computer vision problem.

## 1 Introduction

Stereo-vision is by far the most popular method to solve the problem of three-dimensional reconstruction. Inspired by biological vision, its uses two ore more images taken from different viewpoints in order to extract depth information. This field of computer vision has been under investigation for many years with a thriving literature and algorithms on the subject[17, 8]. Thought the major part of the existing algorithms follow the classical processing (segmentation, feature extraction and correspondence), new algorithms have been introduced that propose novel ways to solve the problem such as voxel coloring[19] and space carving[13, 12]. These methods are based on computations in the three-dimensional space constructing volumes and surfaces that are consistent with the input images[6]. In broad outline, these approaches start by an initial virtual volume comprised of voxels in the scene from which voxels that are not part of real objects are removed iteratively leaving at the end a volume that represent the scene.

The fly algorithm[14] is an evolutionary computation method applied to fast three-dimensional reconstruction. It belongs to the above class of methods since it operates in the three-dimensional space; however, unlike the above-mentioned algorithms which explore the search space exhaustively (each voxel is checked), the fly algorithm uses an evolutionary approach to explores the space and only interesting regions are explored. This reduces the size of the search space and the reconstruction problem is formulated as an optimization problem searching for the best representation of

the scene. The algorithm has been applied to different problems where fast three-dimensional reconstruction is needed. Such problems includes obstacle detection[2], robot navigation[3] and tomographic image interpretation[4].

In this work we focus on the running time and the quality of the reconstructions by comparing the algorithm’s result with established benchmark data. We will begin by brief description of the algorithm, after which we present some experimental results, comment on them and finally conclude and present future direction of research we are interested to pursue.

## 2 The algorithm

We choose here to not go into much details in the description of the algorithm, we will only describe the main ideas, details can be found in [14, 2, 3].

The Fly algorithm is an evolutionary three-dimensional reconstruction method based on the evolution of a population of 3-D points (the flies) in space. The algorithm evolves the fly population, first initialized without any reference to the scene, optimizing a fitness function designed such that the flies converge onto the physical objects in the scene. The result of the reconstruction is the population of the flies (fig. 1). Each fly’s genome is a real vector  $(x, y, z)$  that represent the position of a fly in space. The population evolves using classical genetic operator[2] (Gaussian mutation, barycentric crossover and fitness sharing).

Unlike classical evolutionary algorithms in which each genome represents a complete potential solutions, and in which the solution of the problem is the best fit genome, the fly algorithm follows a different scheme where the solution is distributed on the population: each fly is part of the solution to the problem and the entire population is the complete solution[5]. The

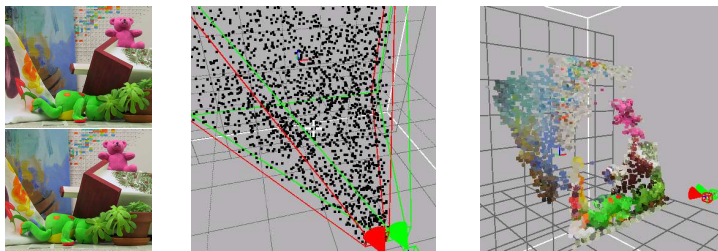


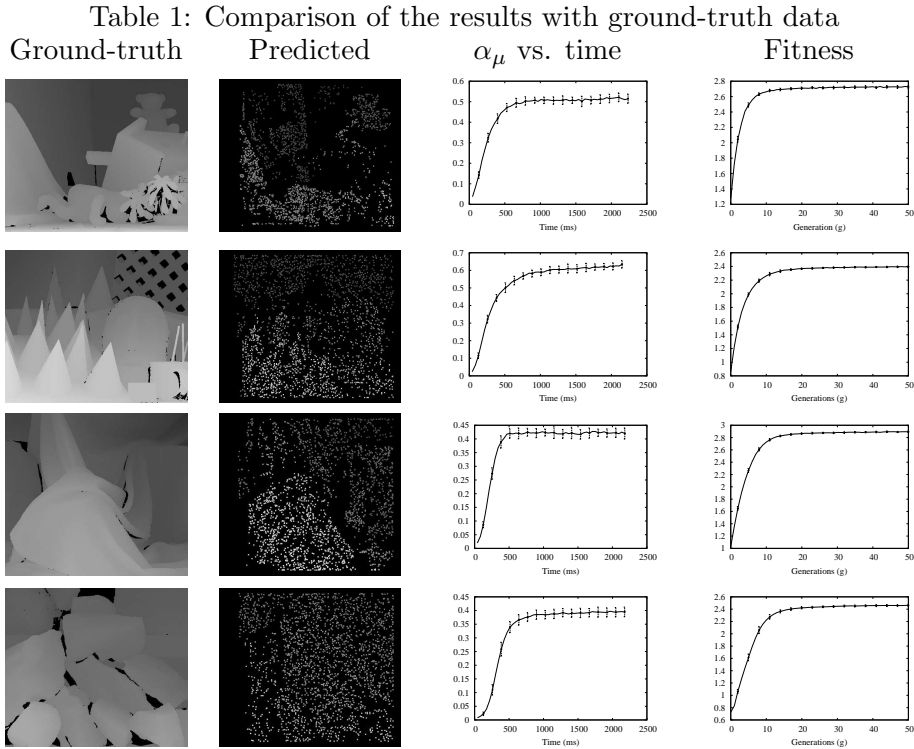
Figure 1: Input stereo images (left), the initialization of the fly population (middle) and the reconstruction result of the fly algorithm (right).

fitness function of a fly, is based on the similarity between the neighborhoods around the fly’s projections on the images. The idea behind this function relies on the fact that, if a fly is on the surface of an opaque object, the neighborhood around its projections in both images will normally be highly similar. Conversely, if the fly is not on the surface of an object, then these

neighborhoods will be poorly correlated. Therefore, the fitness value of a fly measures the similarity of projections<sup>1</sup>. Since the aim of the algorithm is to find a three-dimensional representation of the scene, it is thus important for the population to cover the scene. Thus to avoid crowds, a sharing procedure [16] reduces the fitness values of flies located in crowded areas. The selection pressure drives these flies out of population creating new ones in less crowded areas. Selection in the evolutionary process leads to populations with better fit flies and by the same way better representations of the scene.

### 3 Experiments

The experiment presented bellow aim at measuring the quality and the precision of the results given by algorithm. These results are mean values over 50 runs of the algorithm with the same parameters (population size, sharing coefficient[14] and mutation strength). We will asses the accuracy of the fly algorithm’s results using the benchmark set of images provided by the the “Middlebury Stereo Vision Page[10]”. The image set is provided along with ground-truth data, in the form of disparity maps[8]. We base our



<sup>1</sup>The calibration parameter are assumed known so that the projections could be computed using projective geometry.

performance analysis on the comparisons of the disparity maps produced by the fly algorithm with the ground-truth disparity maps. Examples of such maps can be seen on (tab. 1). In order to compare the disparity results we measure the proportion of flies, with the correct disparity.

The curves of tab. 1 show the proportion of flies ( $\alpha_\mu$ ), out of the entire population (1024 flies), with the correct disparity as a function of time. This quantity increase with time which shows that the reconstruction is refined as the evolution goes. We notice however, that the curves stabilizes at some point in time, near 500 ms in these experiments, and that not much progress is gained afterwards. The 500 ms mark corresponds to 10 generations of evolution for a population size of 1024. The fitness curves on the right represent the evolution of the average fitness of the population with respect to generations. Here again we notice the “diminishing return”, since starting a generation 10 the progress is not noticeable. This behavior is typical for evolutionary computation algorithms, reflecting the fact that at an early stage the algorithm is in its exploration phase, whereas later on it enters an exploitation phase, refining its search around the optima found.

In the first  $\alpha_\mu$  curve, the progress stops at 0.51 stating that 51% of the population have the true disparity, which may seem not enough. It is however, worth to note that the ground-truth disparities were obtained with sub-pixel accuracy which is not the case for our algorithm. If we allow an error margin of 1, in other words, measure the proportion of flies that are within 1 of the true disparity, then  $\alpha_\mu$  jumps to 0.9 and higher. Figure 2 illustrates this showing  $\alpha_\mu$  for different error tolerances. We believe thought still under investigation, that using a sub-pixel accuracy may give better results. However, this potential gain will be at the expense of computation time.

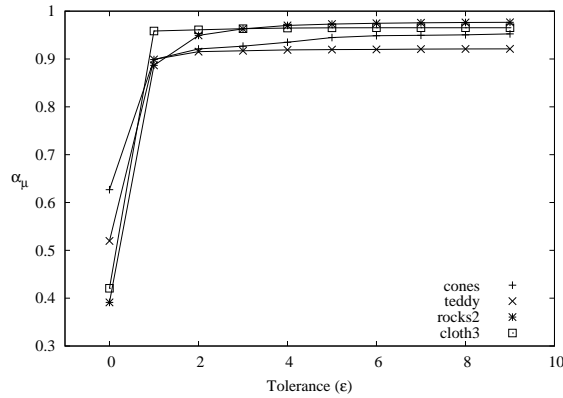


Figure 2: The proportion of flies with disparities within  $\epsilon$  error from the true disparities.

## 4 Discussion and conclusions

In this work, we have analyzed the quality of the results provided by the fly algorithm using benchmark images. The comparison of the results with ground-truth data shows that the quality of the 3D reconstructions of the fly algorithm falls within a good range of the true reconstruction.

Artificial evolutionary algorithms have become widely used optimization techniques and we are starting to see many application of evolutionary algorithms in computer vision problems[18, 15, 7, 1, 9]. Although, misconceptions about these algorithms, remain: they are thug to be slow and not well suited for time-critical applications.

The best ranked algorithm[11] on the Middlebury benchmark reports running times of 14-25 sec to produce a reconstruction. We have shown that on the same images, the fly algorithm is capable to produce “good enough” results in a very short time (0.5 sec). Thought these results may not be well suited for certain type of applications where accurate reconstructions are needed, they are however sufficient for applications such as obstacle detection and robot navigation. The use of more elaborate methods, such as sub-pixel interpolation, may increase the accuracy of the algorithm, but will surely increase the running time since the computation will be more complex.

This work shows that a well designed evolutionary algorithm may be well suited for applications in three-dimensional reconstruction, and we hope to through this to motivate further the use of methods such as evolutionary computation in image analysis.

## References

- [1] G. Bebis, S. Louis, Y. Varol, and A. Yfantis. Genetic object recognition using combinations of views. *IEEE Transactions on Evolutionary Computation*, 6(2):132–146, April 2002.
- [2] A. Boumaza and J. Louchet. Dynamic flies: Using real-time parisian evolution in robotics. In E. J. W. Boers et al, editor, *Applications of Evolutionary Computing*, volume 2037 of *LNCS*, pages 288–297. Springer-Verlag, April 2001.
- [3] A. Boumaza and J. Louchet. Mobile robot sensor fusion using flies. In G. R. Raidl et al, editor, *Applications of Evolutionary Computing*, volume 2611 of *LNCS*, pages 357–367. Springer-Verlag, April 2003.
- [4] A. Bousquet and J-M. Rocchisani J. Louchet. Fully three-dimensional tomographic evolutionary reconstruction in nuclear medicine. In *Proceeding of the 8<sup>th</sup> International Conference on Artificial Evolution EA’07*.

- [5] P. Collet, E. Lutton, F. Raynal, and M. Schoenauer. Individual gp: an alternative viewpoint for the resolution of complex problems. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honovar, M. Jakiela, and R. E. Smith, editors, *Genetic and Evolutionary Computation Conference GECCO99*. Morgan Kaufmann, San Francisco, CA, 1999.
- [6] C. Dyer. *Volumetric Scene Reconstruction from Multiple Views*, pages 469–489. Kluwer, Boston, 2001.
- [7] Yong Fan, Tianzi Jiang, and David J. Evans. Volumetric segmentation of brain images using parallel genetic algorithms. *IEEE Transactions on Medical Imaging*, 21(8), 2002.
- [8] O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometrical Viewpoint*. The MIT Press, Cambridge, MA, 1993.
- [9] Yoshi Fujiwara and Hidefumi Sawai. Evolutionary computation applied to mesh optimization of a 3-d facial image. *IEEE Transactions on Evolutionary Computation*, 3(2):113–123, 1999.
- [10] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN,, June 2007.
- [11] A. Klaus, M Sormann, and K Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, pages 15–18, Washington, DC, USA, 2006. IEEE Computer Society.
- [12] K. N. Kutulakos. Approximate n-view stereo. In *Proc. of the European Conference on Computer Vision*, volume LNCS 1842, pages 67–83. Springer - Verlag, 2000.
- [13] K. N. Kutulakos and S. M. Seits. A theory of shape by space carving. *Journal of computer vision*, 38(3):199–218, 2000.
- [14] J. Louchet, M. Guyon, M. Lesot, and A. Boumazza. Dynamic flies: a new pattern recognition tool applied to stereo sequence processing. *Pattern recognition letters*, 23:335–345, 2002.
- [15] E. Lutton and P. Martinez. A genetic algorithm for the detection of d geometric primitives in images. In *The Proceedings of the International Conference on Pattern Recognition, ICPR'94*, pages 526–528, Los Alamitos, CA, October 9-13 1994. IEEE Computer Society.

- [16] Samir W. Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, 1995.
- [17] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.
- [18] G. Roth and M. D. Levine. Geometric primitive extraction using genetic algorithm. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society Press, 1992.
- [19] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *Journal of computer vision*, 38(2):151–173, 1999.